Linear Algebra and its Applications 434 (2011) 2244–2268

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Synthetic boundary conditions for image deblurring☆

Ying Wai (Daniel) Fan, James G. Nagy*

Mathematics and Computer Science, Emory University, Atlanta, GA, USA

ARTICLE INFO

Article history:
Received 24 July 2009
Accepted 19 November 2009
Available online 25 January 2010

Submitted by V. Mehrmann

ABSTRACT

In this paper we introduce a new boundary condition that can be used when reconstructing an image from observed blurred and noisy data. Our approach uses information from the observed image to enforce boundary conditions that continue image features such as edges and texture across the boundary. Because of its similarity to methods used in texture synthesis, we call our approach *synthetic boundary conditions*. We provide an efficient algorithm for implementing the new boundary condition, and provide a linear algebraic framework for the approach that puts it in the context of more classical and well known image boundary conditions, including zero, periodic, reflective, and anti-reflective. Extensive numerical experiments show that our new synthetic boundary conditions provide a more accurate approximation of the true image scene outside the image boundary, and thus allow for better reconstructions of the unknown, true image scene.

1. Introduction

The use of advanced imaging technologies is an integral part of scientific research, especially in fields such as biology, medicine and astronomy. Imaging is also an important component of modern security systems (e.g., video surveillance and biometric scanning), and is used to inspect machine parts (e.g, jet engine turbine blades) for possible small, but critical, defects. Although physical limitations of imaging devices, as well as environmental effects, impede the ability to obtain perfect images, the resolution can often be improved through computational postprocessing techniques.

☆ Research supported in part by the NSF under grant DMS-0811031.
* Corresponding author.
E-mail addresses: yfan@mathcs.emory.edu (Y.W. Fan), nagy@mathcs.emory.edu (J.G. Nagy).

0024-3795/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2009.12.021

In this paper we consider the particular, and commonly used, postprocessing technique of image deblurring with a spatially invariant blurring operator (i.e., deconvolution). The image formation process is modeled as a linear inverse problem,

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \boldsymbol{\varepsilon}. \tag{1.1}$$

In the discrete setting, the vector $\mathbf{f}$ contains pixel values of the true (unknown) image scene on a bounded domain. The matrix $\mathbf{A}$ models the distortion (e.g., blurring) in the observed image $\mathbf{g}$, and $\boldsymbol{\varepsilon}$ is additive noise. The underlying continuous mathematical model is often ill-posed, resulting in a discrete problem given by Eq. (1.1) where the matrix $\mathbf{A}$ is severely ill-conditioned, with singular values decaying to zero, without a significant gap to indicate numerical rank. The postprocessing problem is, given $\mathbf{A}$ and $\mathbf{g}$, compute an approximation of $\mathbf{f}$.

The matrix $\mathbf{A}$ is defined by a given *point spread function* (PSF). Algorithms to solve (1.1) exploit structure of $\mathbf{A}$, which depends on the PSF and on the imposed boundary conditions. In general,

$$\mathbf{A} = \mathbf{T} + \mathbf{B} \tag{1.2}$$

where $\mathbf{T}$ has a Toeplitz structure and $\mathbf{B}$, which is defined by the boundary conditions, is often structured, sparse, and low rank.

Boundary conditions make assumptions about how the image behaves outside the field of view, and they are often chosen for algebraic and computational convenience. For example, periodic boundary conditions result in a matrix $\mathbf{A}$ that has a circulant structure, which is diagonalized by the unitary discrete Fourier transform matrix [4]. It is well known that computations with such matrices can be done very efficiently by using fast Fourier transforms (FFT) [16]. Note that periodic boundary conditions assume that the true infinite scene can be represented as a mosaic of a single finite dimensional image, repeated periodically in all directions. Thus, although computationally convenient, for most images it is difficult to provide a physical justification for the use of periodic boundary conditions.

Other boundary conditions can have better physical justification. For example, if the image is assumed to have a black background (such as in the case of astronomical images), then zero boundary conditions may provide a good physical representation for the image scene outside the viewable region. In this case $\mathbf{B}$ is zero, and thus $\mathbf{A}$ has a Toeplitz structure. Although direct filtering type methods cannot be implemented as efficiently as in the case of circulant structures, it is possible to effectively use iterative methods [15,17]. However, if there are significant features near the image boundary, then zero boundary conditions may not provide a physically accurate model of the infinite scene.

If there are significant features that overlap the edge of the viewable region, then it may make sense to use reflective boundary conditions, where it is assumed that the scene outside the viewable region is a mirror reflection of the scene inside the viewable region [19]. In this case the matrix $\mathbf{A}$ has a Toeplitz-plus-Hankel structure. Iterative methods for such matrices can be implemented efficiently, and, moreover, if the PSF satisfies a strong symmetry condition, $\mathbf{A}$ can be diagonalized by the orthogonal discrete cosine transformation matrix, and spectral filtering methods can be implemented very efficiently [14]. With reflective boundary conditions, continuity of the graylevel values of the image is maintained.

More recently, anti-reflective boundary conditions have been proposed, which extend the pixel values across the boundary in such a way that continuity of the image and of the normal derivative are preserved at the boundary [6,7,20]. In this case the structure of $\mathbf{A}$ is Toeplitz-plus-Hankel, plus an additional structured low rank matrix. As with reflective boundary conditions, iterative methods for such matrices can be implemented efficiently, and, moreover, if the PSF satisfies a strong symmetry condition, spectral filtering methods can be implemented very efficiently (though the details are a bit more complicated); see [1] for more details.

In this paper we propose a new approach, which we call synthetic boundary conditions. Our goal is to not necessarily continue graylevels at the boundary, but instead to develop a scheme that can continue edge directions and textures of the image inside the viewable region to outside the image boundary. We remark that, although our discussion is for the specific problem of image deblurring (deconvolution), the approach we propose in this paper can be used in other imaging applications as well.

This paper is outlined as follows. In Section 2 we briefly review the most commonly used boundary conditions (zero, periodic, reflective, and anti-reflective) for image deblurring, and introduce our new synthetic boundary conditions. Using a linear algebraic framework we establish connections between various boundary conditions, and discuss efficient implementation details. The linear algebraic framework is also exploited in Section 3 to construct preconditioners for iterative image deblurring algorithms. In Section 4 we provide extensive numerical examples to illustrate the effectiveness of the synthetic boundary conditions, and Section 5 contains some concluding remarks.

2. Image deblurring and boundary conditions

In this section we review some classical boundary conditions that are commonly used in imaging deblurring. We illustrate that in each case the matrix $\mathbf{A}$ in Eq. (1.1) can be put in the form given by Eq. (1.2). Since our focus in this section is on the matrix $\mathbf{A}$ and the structure it exhibits when using various boundary conditions, without loss of generality we can assume there is no additive noise in the image formation process, that is $\boldsymbol{\varepsilon} = \mathbf{0}$. The noise will be accounted through regularization methods in the numerical results section.

To simplify the discussion, we begin by describing matrix structures for one dimensional problems. We then extend the discussion to two dimensional problems. Finally we propose a new approach that uses information from the observed image to enforce continuity of image features such as edges and texture across the boundary.

2.1. One dimensional problems

We begin with the one dimensional problem because the matrix descriptions are easier to follow. The two dimensional problem is discussed in the next subsection. We use notation similar to that in [19]. Suppose $\mathbf{g}$ is a one dimensional image (i.e., a signal) that is obtained by convolving the PSF $\mathbf{h}$ with (an unknown, true) signal $\mathbf{f}_{\text{true}}$, where

$$\mathbf{g} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{bmatrix} \quad \text{and} \quad \mathbf{h} = \begin{bmatrix} h_{-m} \\ \vdots \\ h_{-1} \\ h_0 \\ h_1 \\ \vdots \\ h_m \end{bmatrix}$$

and $n \geqslant 2m+1$. Then the convolution model implies that for each $k = 0, 1, \ldots, n-1$,

$$g_k = \sum_{i=-m}^{m} h_i f_{k-i},$$

which can be written in matrix–vector form as

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{bmatrix} = \begin{bmatrix} h_m & \cdots & h_0 & \cdots & h_{-m} & & & & & & \\ & \ddots & \vdots & \ddots & \vdots & \ddots & & & & & \\ & & h_m & & h_0 & & h_{-m} & & & & \\ & & & \ddots & \vdots & \ddots & \vdots & \ddots & & & \\ & & & & h_m & & h_0 & & h_{-m} & & \\ & & & & & \ddots & \vdots & \ddots & \vdots & \ddots & \\ & & & & & & h_m & \cdots & h_0 & \cdots & h_{-m} \end{bmatrix}$$

$$\begin{bmatrix} f_{-m} \\ \vdots \\ f_{-1} \\ \hline f_0 \\ \vdots \\ f_{n-1} \\ \hline f_n \\ \vdots \\ f_{n-1+m} \end{bmatrix} \tag{2.1}$$

where we use horizontal lines in $\mathbf{f}_{\text{true}}$ to denote the boundaries of the field of view in the true image scene, which correspond to those of the observed signal $\mathbf{g}$. Although m is generally small compared to n, the problem is underdetermined since values of $\mathbf{g}$ near the boundary (such as g_0 and g_{n-1}) depend on values of $\mathbf{f}_{\text{true}}$ outside the field of view. It will be convenient to rewrite Eq. (2.1) as

$$\mathbf{g} = \left[\mathbf{T}_{-1} \;\middle|\; \mathbf{T} \;\middle|\; \mathbf{T}_1\right] \left[\begin{array}{c} \mathbf{f}_{-1} \\ \hline \mathbf{f} \\ \hline \mathbf{f}_1 \end{array}\right],$$

where $\mathbf{T}_{-1}$, $\mathbf{T}$ and $\mathbf{T}_1$ are the following Toeplitz matrices

$$\overbrace{\begin{bmatrix} h_m & \cdots & h_1 \\ & \ddots & \vdots \\ & & h_m \\ & & \\ & & \\ & & \\ & & \\ & & \end{bmatrix}}^{\mathbf{T}_{-1}} \overbrace{\begin{bmatrix} h_0 & \cdots & \cdots & h_{-m} & & & & \\ \vdots & \ddots & & \vdots & \ddots & & & \\ \vdots & & \ddots & \vdots & & h_{-m} & & \\ h_m & & & h_0 & & \vdots & \ddots & \\ & \ddots & & \vdots & \ddots & \vdots & & \ddots \\ & & \ddots & \vdots & & h_0 & & h_{-m} \\ & & & h_m & & \vdots & \ddots & \vdots \\ & & & & \ddots & \vdots & & \vdots \\ & & & & & h_m & \cdots & h_0 \end{bmatrix}}^{\mathbf{T}} \overbrace{\begin{bmatrix} & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ h_{-m} & & \\ \vdots & \ddots & \\ h_{-1} & \cdots & h_{-m} \end{bmatrix}}^{\mathbf{T}_1} \tag{2.2}$$

and

$$\mathbf{f}_{-1} = \begin{bmatrix} f_{-m} \\ \vdots \\ f_{-1} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_0 \\ \vdots \\ f_{n-1} \end{bmatrix}, \quad \mathbf{f}_1 = \begin{bmatrix} f_n \\ \vdots \\ f_{n-1+m} \end{bmatrix}.$$

Since $\mathbf{f}_{-1}$ and $\mathbf{f}_1$ are outside the field of view, and are therefore not measurable, boundary conditions replace these with values that can be either set *a priori* or obtained from information within the field of view. Specifically, $\mathbf{f}_{-1}$ and $\mathbf{f}_1$ are replaced with

$$\hat{\mathbf{f}}_{-1} = \mathbf{S}_{-1}\mathbf{f} \quad \text{and} \quad \hat{\mathbf{f}}_1 = \mathbf{S}_1\mathbf{f},$$

where $\mathbf{S}_{-1}$ and $\mathbf{S}_1$ are matrices defined by the boundary conditions (specific examples are given below). With this notation, and with $\boldsymbol{\varepsilon} = \mathbf{0}$, Eq. (2.1) is approximated by

$$\mathbf{g} = \mathbf{A}\mathbf{f}, \tag{2.3}$$

where $\mathbf{A} = \mathbf{T} + \mathbf{B}$ and $\mathbf{B} = \mathbf{T}_{-1}\mathbf{S}_{-1} + \mathbf{T}_1\mathbf{S}_1$. Some well-known examples include:

- For *zero* boundary conditions it is assumed that the signal is always zero outside the field of view; that is, $\hat{\mathbf{f}}_{-1} = \hat{\mathbf{f}}_1 = \mathbf{0}$. In this case, $\mathbf{S}_{-1} = \mathbf{S}_1 = \mathbf{O}$, where $\mathbf{O}$ is a matrix of all zeros. Thus $\mathbf{B} = \mathbf{O}$, and $\mathbf{A} = \mathbf{T}$.
- For *periodic* boundary conditions we use

$$\hat{\mathbf{f}}_{-1} = \begin{bmatrix} f_{n-m} \\ f_{n-m+1} \\ \vdots \\ f_{n-1} \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{f}}_1 = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{m-1} \end{bmatrix}.$$

  Thus, $\mathbf{S}_{-1} = [\mathbf{O} \quad \mathbf{I}]$ and $\mathbf{S}_1 = [\mathbf{I} \quad \mathbf{O}]$, where $\mathbf{O}$ is a matrix of all zeros, and $\mathbf{I}$ is an $m \times m$ identity matrix. In this case, $\mathbf{B} = [\mathbf{O} \quad \mathbf{T}_{-1}] + [\mathbf{T}_1 \quad \mathbf{O}]$, and it is not difficult to show that $\mathbf{A} = \mathbf{T} + \mathbf{B}$ is a circulant matrix. Note that the rank$(\mathbf{B}) = 2m$, which is (often much) less than n.
- For *reflective* boundary conditions we use

$$\hat{\mathbf{f}}_{-1} = \begin{bmatrix} f_{m-1} \\ \vdots \\ f_1 \\ f_0 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{f}}_1 = \begin{bmatrix} f_{n-1} \\ \vdots \\ f_{n-m+1} \\ f_{n-m} \end{bmatrix}.$$

  Thus, $\mathbf{S}_{-1} = [\mathbf{O} \quad \mathbf{I}]\mathbf{J}$ and $\mathbf{S}_1 = [\mathbf{I} \quad \mathbf{O}]\mathbf{J}$, where $\mathbf{J}$ is the "reversal" permutation matrix,

$$\mathbf{J} = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

  In this case, $\mathbf{B} = [\mathbf{O} \quad \mathbf{T}_{-1}]\mathbf{J} + [\mathbf{T}_1 \quad \mathbf{O}]\mathbf{J}$ is a Hankel matrix, and so $\mathbf{A} = \mathbf{T} + \mathbf{B}$ is a Toeplitz-plus-Hankel matrix. Again we see that the rank$(\mathbf{B}) = 2m$.
- For *anti-reflective* boundary conditions, originally proposed by Serra-Capizzano [20], we use

$$\hat{\mathbf{f}}_{-1} = \begin{bmatrix} 2f_0 - f_m \\ \vdots \\ 2f_0 - f_2 \\ 2f_0 - f_1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{f}}_1 = \begin{bmatrix} 2f_{n-1} - f_{n-2} \\ \vdots \\ 2f_{n-1} - f_{n-m} \\ 2f_{n-1} - f_{n-m-1} \end{bmatrix}.$$

  Thus, $\mathbf{S}_{-1} = [\mathbf{O} \quad -\mathbf{I} \quad 2\mathbf{e}]\mathbf{J}$ and $\mathbf{S}_1 = [2\mathbf{e} \quad -\mathbf{I} \quad \mathbf{O}]\mathbf{J}$, where $\mathbf{e}$ is a vector of ones. In this case $\mathbf{B} = [\mathbf{O} \quad -\mathbf{T}_{-1} \quad 2\mathbf{T}_{-1}\mathbf{e}]\mathbf{J} + [2\mathbf{T}_1\mathbf{e} \quad -\mathbf{T}_1 \quad \mathbf{O}]\mathbf{J}$ is the sum of a Hankel matrix and a matrix with rank equal to two. The matrix $\mathbf{A} = \mathbf{T} + \mathbf{B}$ is then Toeplitz-plus-Hankel, plus an additional rank-2 matrix. Note that in this case the rank$(\mathbf{B}) = 2m + 2$.

Observe that in all of the above examples, the noise-free one dimensional deblurring problem can be represented as

$$\mathbf{g} = \mathbf{A}\mathbf{f}, \quad \mathbf{A} = \mathbf{T} + \mathbf{B}$$

where $\mathbf{T}$ is a Toeplitz matrix, and $\mathbf{B}$ is a matrix defined by the boundary condition, which is structured, and if $m \ll n$, also sparse and low rank. This linear algebra formulation can be extended to higher dimensions.

2.2. Two dimensional problems

Extending this linear algebraic formulation to two dimensional imaging problems is not so difficult, but the notation can be a bit cumbersome. To facilitate readability, we assume all images are square (e.g., $n \times n$) arrays of pixel values, and that the PSF is separable.

Suppose that $\mathbf{G}$ is an $n \times n$ image that is obtained by convolving the $m \times m$ PSF $\mathbf{H}$ with (an unknown, true) image $\mathbf{F}_{\text{true}}$, where $n \geqslant 2m+1$. Then the convolution model implies that for each $k, \ell = 0, 1, \ldots, n-1$,

$$g_{k,\ell} = \sum_{i=-m}^{m} \sum_{j=-m}^{m} h_{ij} f_{k-i,\ell-j}. \tag{2.4}$$

If the PSF is separable (i.e., the vertical blurring operation is independent of the horizontal blurring operation), then there are vectors

$$\mathbf{h}_c = \begin{bmatrix} h_{-m}^{(c)} \\ \vdots \\ h_{-1}^{(c)} \\ h_0^{(c)} \\ h_1^{(c)} \\ \vdots \\ h_m^{(c)} \end{bmatrix} \quad \text{and} \quad \mathbf{h}_r = \begin{bmatrix} h_{-m}^{(r)} \\ \vdots \\ h_{-1}^{(r)} \\ h_0^{(r)} \\ h_1^{(r)} \\ \vdots \\ h_m^{(r)} \end{bmatrix}$$

such that

$$\mathbf{H} = \mathbf{h}_c \mathbf{h}_r^T \quad \Leftrightarrow \quad h_{ij} = h_i^{(c)} h_j^{(r)},$$

where $\mathbf{h}_c$ and $\mathbf{h}_r$ represent, respectively, the vertical and horizontal components of the PSF [14]. In this case, the convolution equation (2.4) becomes

$$\begin{aligned} g_{k,\ell} &= \sum_{i=-m}^{m} \sum_{j=-m}^{m} h_{ij} f_{k-i,\ell-j} \\ &= \sum_{i=-m}^{m} \sum_{j=-m}^{m} h_i^{(c)} h_j^{(r)} f_{k-i,\ell-j} \\ &= \sum_{i=-m}^{m} \left(h_i^{(c)} \sum_{j=-m}^{m} \left(f_{k-i,\ell-j} h_j^{(r)} \right) \right), \end{aligned}$$

which can be written in matrix–vector form as

$$\mathbf{G} = \begin{bmatrix} \mathbf{T}_{c,-1} & \mathbf{T}_c & \mathbf{T}_{c,1} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{-1,-1} & \mathbf{F}_{-1,0} & \mathbf{F}_{-1,1} \\ \mathbf{F}_{0,-1} & \mathbf{F} & \mathbf{F}_{0,1} \\ \mathbf{F}_{1,-1} & \mathbf{F}_{1,0} & \mathbf{F}_{1,1} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{r,-1}^T \\ \mathbf{T}_r^T \\ \mathbf{T}_{r,1}^T \end{bmatrix}, \tag{2.5}$$

where $[\mathbf{T}_{c,-1} \quad \mathbf{T}_c \quad \mathbf{T}_{c,1}]$ and $[\mathbf{T}_{r,-1} \quad \mathbf{T}_r \quad \mathbf{T}_{r,1}]$ are identical in structure to the matrices given in Eq. (2.2), $\mathbf{F}$ is the $n \times n$ portion of the true image scene within the field of view (defined by $\mathbf{G}$), and $\mathbf{F}_{i,j}$ represent sections of the scene that are outside the field of view. As in the one dimensional model, since $\mathbf{F}_{i,j}$ are outside the field of view, we use boundary conditions to replace these with values that are either set *a priori* (e.g., to zero), or with values that can be obtained from information within the field of view. That is, the array representing the true image scene

$$\mathbf{F}_{\text{true}} = \begin{bmatrix} \mathbf{F}_{-1,-1} & \mathbf{F}_{-1,0} & \mathbf{F}_{-1,1} \\ \mathbf{F}_{0,-1} & \mathbf{F} & \mathbf{F}_{0,1} \\ \mathbf{F}_{1,-1} & \mathbf{F}_{1,0} & \mathbf{F}_{1,1} \end{bmatrix}$$

is replaced with

$$\begin{bmatrix} \widehat{\mathbf{F}}_{-1,-1} & \widehat{\mathbf{F}}_{-1,0} & \widehat{\mathbf{F}}_{-1,1} \\ \widehat{\mathbf{F}}_{0,-1} & \mathbf{F} & \widehat{\mathbf{F}}_{0,1} \\ \widehat{\mathbf{F}}_{1,-1} & \widehat{\mathbf{F}}_{1,0} & \widehat{\mathbf{F}}_{1,1} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{c,-1}\mathbf{F}\mathbf{S}_{r,-1}^T & \mathbf{S}_{c,-1}\mathbf{F} & \mathbf{S}_{c,-1}\mathbf{F}\mathbf{S}_{r,1}^T \\ \mathbf{F}\mathbf{S}_{r,-1}^T & \mathbf{F} & \mathbf{F}\mathbf{S}_{r,1}^T \\ \mathbf{S}_{c,1}\mathbf{F}\mathbf{S}_{r,-1}^T & \mathbf{S}_{c,1}\mathbf{F} & \mathbf{S}_{c,1}\mathbf{F}\mathbf{S}_{r,1}^T \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{S}_{c,-1} \\ \mathbf{I} \\ \mathbf{S}_{c,1} \end{bmatrix} \mathbf{F} \begin{bmatrix} \mathbf{S}_{r,-1}^T & \mathbf{I} & \mathbf{S}_{r,1}^T \end{bmatrix},$$

where $\mathbf{S}_{c,-1}$ and $\mathbf{S}_{c,1}$ define the vertical boundary conditions (i.e., those imposed at the top and bottom of the image), and $\mathbf{S}_{r,-1}$ and $\mathbf{S}_{r,1}$ define the horizontal boundary conditions (i.e., those imposed at the left and right of the image).

We remark that our approach to defining boundary conditions does not require a separable PSF. However, if the blur is separable, then Eq. (2.5) can be approximated with

$$\mathbf{G} = \begin{bmatrix} \mathbf{T}_{c,-1} & \mathbf{T}_c & \mathbf{T}_{c,1} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{c,-1} \\ \mathbf{I} \\ \mathbf{S}_{c,1} \end{bmatrix} \mathbf{F} \begin{bmatrix} \mathbf{S}_{r,-1}^T & \mathbf{I} & \mathbf{S}_{r,1}^T \end{bmatrix} \begin{bmatrix} \mathbf{T}_{r,-1}^T \\ \mathbf{T}_r^T \\ \mathbf{T}_{r,1}^T \end{bmatrix}$$
$$= \left(\mathbf{T}_{c,-1}\mathbf{S}_{c,-1} + \mathbf{T}_c + \mathbf{T}_{c,1}\mathbf{S}_{c,1}\right) \mathbf{F} \left(\mathbf{S}_{r,-1}^T\mathbf{T}_{r,-1}^T + \mathbf{T}_r^T + \mathbf{S}_{r,1}^T\mathbf{T}_{r,1}^T\right),$$

or, equivalently, we can write this in matrix–vector form as

$$\begin{aligned} \mathbf{g} &= \left(\left(\mathbf{T}_{r,-1}\mathbf{S}_{r,-1} + \mathbf{T}_r + \mathbf{T}_{r,1}\mathbf{S}_{r,1}\right) \otimes \left(\mathbf{T}_{c,-1}\mathbf{S}_{c,-1} + \mathbf{T}_c + \mathbf{T}_{c,1}\mathbf{S}_{c,1}\right)\right)\mathbf{f} \\ &= \left(\mathbf{T}_r \otimes \mathbf{T}_c + \mathbf{T}_r \otimes \left(\mathbf{T}_{c,-1}\mathbf{S}_{c,-1} + \mathbf{T}_{c,1}\mathbf{S}_{c,1}\right)\right. \\ &\quad \left. + \left(\mathbf{T}_{r,-1}\mathbf{S}_{r,-1} + \mathbf{T}_{r,1}\mathbf{S}_{r,1}\right) \otimes \left(\mathbf{T}_{c,-1}\mathbf{S}_{c,-1} + \mathbf{T}_c + \mathbf{T}_{c,1}\mathbf{S}_{c,1}\right)\right)\mathbf{f} \end{aligned}$$

where $\otimes$ denotes Kronecker product, and $\mathbf{g} = \mathrm{vec}(\mathbf{G})$ and $\mathbf{f} = \mathrm{vec}(\mathbf{F})$. Again we see that the (noise-free) image deblurring problem with spatially invariant, separable blur, can be represented as

$$\mathbf{g} = \mathbf{A}\mathbf{f}, \quad \mathbf{A} = \mathbf{T} + \mathbf{B}$$

where $\mathbf{T} = \mathbf{T}_r \otimes \mathbf{T}_c$ is a block Toeplitz matrix with Toeplitz blocks (BTTB), and $\mathbf{B} = \mathbf{T}_r \otimes (\mathbf{T}_{c,-1}\mathbf{S}_{c,-1} + \mathbf{T}_{c,1}\mathbf{S}_{c,1}) + (\mathbf{T}_{r,-1}\mathbf{S}_{r,-1} + \mathbf{T}_{r,1}\mathbf{S}_{r,1}) \otimes (\mathbf{T}_{c,-1}\mathbf{S}_{c,-1} + \mathbf{T}_c + \mathbf{T}_{c,1}\mathbf{S}_{c,1})$ is defined by the boundary conditions. Note that if the blur is not separable, then we do not get neat Kronecker product decompositions of $\mathbf{T}$ and $\mathbf{B}$, but we still get the basic form where $\mathbf{T}$ is BTTB and $\mathbf{B}$ is a structured (and typically sparse) matrix.

All of the boundary conditions discussed in the previous subsection for one dimensional problems extend naturally to the two dimensional problem. For example, in the case of reflective boundary conditions, we use

$$\mathbf{S}_{c,-1} = \mathbf{S}_{r,-1} = \begin{bmatrix} \mathbf{O} & \mathbf{I} \end{bmatrix}\mathbf{J} \quad \text{and} \quad \mathbf{S}_{c,1} = \mathbf{S}_{r,1} = \begin{bmatrix} \mathbf{I} & \mathbf{O} \end{bmatrix}\mathbf{J}.$$

In the next subsection we propose a new boundary condition that is more effective at continuing edges and texture of the image across the boundary than zero, periodic, reflective, and anti-reflective approaches.

2.3. Synthetic boundary conditions

As mentioned in Section 1, it is unlikely that a true image scene would be modeled well by periodic boundary conditions, and zero boundary conditions only make sense for scenes with a black background. There may be some rare cases when reflective and anti-reflective boundary conditions provide a good model of the true image scene outside the field of view. In this paper we develop an approach that provides a more realistic extension of pixels across the boundary. For example, texture and edges should be extended sensibly. The motivation for our approach comes from observing that the problem of defining appropriate boundary conditions is similar to the image recovery problem, in

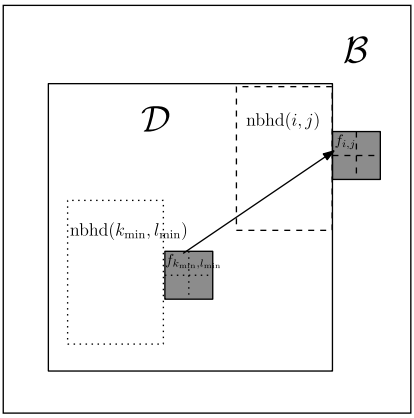


Fig. 2.1. Illustration of how the synthetic boundary condition is determined. Specifically, $(k_{\min}, l_{\min}) = \arg\min_{k,l} \mathrm{SSD}(\mathrm{nbhd}(i,j), \mathrm{nbhd}(k,l))$.

which part of the image is damaged and the aim is to recover missing pixels. In our case, the region we wish to recover corresponds to those pixels outside the boundary. Two common approaches for the image recovery problem are image inpainting [2] and texture synthesis [9]. Image inpainting tries to extend the geometric structure of the image, while texture synthesis extends the texture pattern into the unknown region. In this paper we use the texture synthesis approach.

With the image recovery idea in mind, we wish to determine a relationship between (unknown) pixel values outside the boundary to those pixel values inside the boundary. Using a basic texture synthesis approach, we can try to find a pixel in the viewable region whose neighborhood (e.g., a rectangular region) is most similar to the corresponding neighborhood of the boundary pixel we wish to fill in. If this idea is applied to a blurred image, it can extend edges across the boundary well, but there is little hope that it can also extend the texture, as texture information is lost in blurring. Hence, instead of copying single pixels, we propose to copy small patches that contain the required texture information. This idea is similar to the generalization of texture synthesis to image quilting [8].

To describe more precisely our approach for synthetic boundary conditions, we need a bit of notation. Let

$$\begin{aligned} \mathcal{D} &= [0, n-1] \times [0, n-1] && \text{(domain)},\\ \mathcal{B} &= ([-m, n+m-1] \times [-m, n+m-1]) \backslash \mathcal{D} && \text{(border)}. \end{aligned}$$

Then, using for example 2×2 patches, for each $[i, i+1] \times [j, j+1]$ patch $\in \mathcal{B}$:

- Find

$$(k_{\min}, \ell_{\min}) = \arg\min_{k,\ell} \mathrm{SSD}(\mathrm{nbhd}(i,j), \mathrm{nbhd}(k,\ell)),$$

  where the search is over all $(k,\ell) \in \mathcal{D} \cup \{\text{pixels already processed}\}$, and SSD is the sum of squared differences of pixels in the specified neighborhoods.
- Set the boundary pixels in the 2×2 patch to be

$$\begin{aligned} f_{i,j} &= f_{k_{\min},\ell_{\min}},\\ f_{i+1,j} &= f_{k_{\min}+1,\ell_{\min}},\\ f_{i,j+1} &= f_{k_{\min},\ell_{\min}+1},\\ f_{i+1,j+1} &= f_{k_{\min}+1,\ell_{\min}+1}. \end{aligned}$$

This patch-based texture synthesis idea is illustrated in Fig. 2.1. Larger patches can be used, but we have found that 2×2 patches work well.

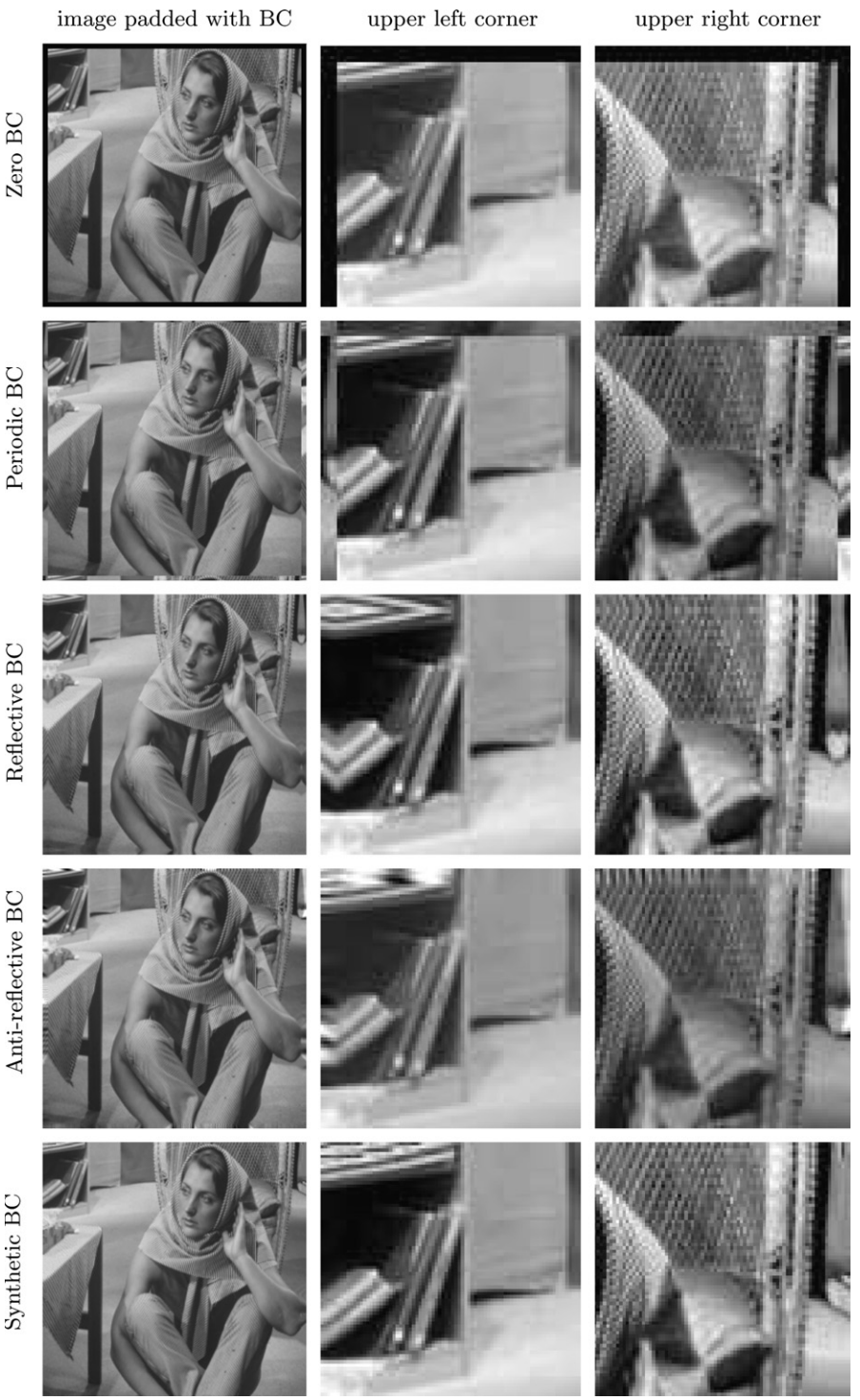


Fig. 2.2. Padded results with different boundary conditions.

An example of padding with synthetic boundary conditions compared to the padding used in other boundary conditions is shown in Fig. 2.2 (the left column shows the full image with padded boundaries, the center column shows a zoom in on the upper left corner, and the right column shows a zoom in on the upper right corner of the image). This figure clearly illustrates that zero and periodic boundary conditions do not preserve continuity of pixel values. Reflective boundary conditions result in continuity of the pixel values across the boundary, but the derivatives of the gray level perpendicular to the image boundary are fixed to be zero. Anti-reflective boundary conditions allow for continuity of the pixel values as well as the derivatives across the boundary. Synthetic boundary conditions do not strive (at least not directly) to maintain continuity, but instead the aim is to match neighborhoods of pixel values. Fig. 2.2 clearly shows that synthetic boundary conditions are much better at extending edges (e.g. of the books in the zoom of the upper left corner) and texture (e.g., of the chair in the zoom of the upper right corner).

The matrix $\mathbf{A}$ for synthetic boundary conditions is similar to the periodic and reflective cases because the pixels in $\mathcal{B}$ are simply copies of pixels in $\mathcal{D}$. Thus, they can be obtained by permutation. To see this, consider again the situation when the blur is separable. Then we can write the matrix–vector model as

$$\mathbf{g} = \left(\begin{bmatrix}\mathbf{T}_{r,-1} & \mathbf{T}_r & \mathbf{T}_{r,1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{T}_{c,-1} & \mathbf{T}_c & \mathbf{T}_{c,1}\end{bmatrix}\right)\mathbf{P}\mathbf{f},$$

where

$$\mathbf{P} = \begin{bmatrix}\mathbf{S}_{r,-1}\\ \mathbf{I}\\ \mathbf{S}_{r,1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{S}_{c,-1}\\ \mathbf{I}\\ \mathbf{S}_{c,1}\end{bmatrix}.$$

Thus, in the case of periodic and reflective boundary conditions, $\mathbf{P}$ is simply a highly structured permutation matrix, which only allows to grab entries from restricted regions of the viewable region. With the use of synthetic boundary conditions we relax the structure of $\mathbf{P}$, and allow the permutation matrix to grab entries more flexibly in the viewable region. The result, as illustrated in Fig. 2.2, is a much better representation of the edges and texture of the image across the boundary.

We emphasize that synthetic boundary conditions are image dependent, and are therefore more capable of extending image features. However, an additional step is needed to estimate the boundary conditions. For efficient implementation, the search for $(k_{\min}, l_{\min})$ can be done only over nearby pixels of (i, j), rather than over the whole image. This makes sense since pixels with similar image features (e.g. edge directions, texture) are usually close to each other. For a fixed size for $\text{nbhd}(i, j)$ and $\text{nbhd}(k, l)$ and a fixed size for the search pool, the cost of enforcing synthetic boundary conditions is proportional to the number of pixels to be filled in the border area, i.e. $O(mn)$ for an image with $m \times n$ pixels. Further computational savings in the implementation, similar to that in [8], can be obtained by reusing intermediate values of $\text{SSD}(\text{nbhd}(i, j), \text{nbhd}(k, l))$. We remark that the cost of obtaining the boundary conditions is negligible compared with that of the subsequent iterative methods to deblur the image.

3. Preconditioners for synthetic boundary conditions

For synthetic boundary conditions, the matrix $\mathbf{A}$ does not have the kind of structure that allows efficient implementation of direct filtering type methods. This is similar to the situation when zero boundary conditions are used, or when reflective and anti-reflective boundary conditions are used with a nonsymmetric PSF. In these situations it is necessary to use iterative methods, such as a conjugate gradient type approach (e.g., CG, MINRES, or LSQR). We remark that for conjugate gradient type methods, the matrix $\mathbf{A}$ need not be formed explicitly, all that is needed is an efficient approach to compute matrix–vector multiplications with $\mathbf{A}$. This can be done by exploiting the structure of $\mathbf{A} = \mathbf{T} + \mathbf{B}$; FFTs can be used to multiply $\mathbf{T}$ by accessing only the PSF, and $\mathbf{B}$ is a sparse matrix. Or, alternatively, FFTs can be used on appropriately padded image arrays. The latter is used in our implementation.

The next issue, then, is to consider preconditioning. Note that for the various boundary conditions considered in this paper, we have:

$$\begin{aligned} \text{Zero BC}: &\quad \mathbf{A}_Z = \mathbf{T} + \mathbf{B}_Z, \\ \text{Periodic BC}: &\quad \mathbf{A}_P = \mathbf{T} + \mathbf{B}_P, \\ \text{Reflective BC}: &\quad \mathbf{A}_R = \mathbf{T} + \mathbf{B}_R, \\ \text{Anti-reflective BC}: &\quad \mathbf{A}_A = \mathbf{T} + \mathbf{B}_A, \\ \text{Synthetic BC}: &\quad \mathbf{A}_S = \mathbf{T} + \mathbf{B}_S. \end{aligned}$$

That is, the matrix structures are very similar, and thus we could consider using, for example, $\mathbf{A}_P$, or a symmetrized version of $\mathbf{A}_R$ as a preconditioner for $\mathbf{A}_S$. The important property we need is that it is possible to efficiently compute the spectral decomposition of the preconditioner. Note that $\mathbf{A}_P$ is the standard, and well studied, choice for preconditioning $\mathbf{A}_Z$; see, for example, [3,17,18].

For synthetic boundary conditions, if the PSF is symmetric, or close to being symmetric, then (the symmetrized) $\mathbf{A}_R$ is likely to be the most effective preconditioner. If the PSF is far from being symmetric, then $\mathbf{A}_P$ may be the best choice. Note that if we use $\mathbf{A}_R$ as the preconditioner for $\mathbf{A}_S$, then

$$\mathbf{A}_S - \mathbf{A}_R = \mathbf{B}_S - \mathbf{B}_R \quad \Rightarrow \quad \mathbf{A}_S\mathbf{A}_R^{-1} = \mathbf{I} + (\mathbf{B}_S - \mathbf{B}_R)\,\mathbf{A}_R^{-1}.$$

If the reflective BC is a good approximation of the synthetic BC, then we expect $\mathbf{B}_S - \mathbf{B}_R$ to have small rank and small norm. Thus $\mathbf{A}_R$ would be a good preconditioner for $\mathbf{A}_S$.

Since the image deblurring problem is extremely ill-conditioned, some care needs to be taken when incorporating preconditioning so that noise in the observed data is not magnified when we solve systems with the preconditioner,

$$\mathbf{A}_R\mathbf{x} = \mathbf{A}_R^T\mathbf{x} = \mathbf{y} \tag{3.1}$$

(the first equality is due to the symmetry of $\mathbf{A}_R$ for a symmetrized PSF). This equation can be solved efficiently using the discrete cosine transform (DCT) [19]. However, since $\mathbf{A}_R$ is usually ill-conditioned, we cannot use it directly as a preconditioner without including regularization [5,12].

In this paper we use Tikhonov regularization [10,11,13,21]. Specifically, the spectral decomposition of $\mathbf{A}_R$ is

$$\mathbf{A}_R = \mathbf{C}^T\mathbf{\Sigma}\mathbf{C},$$

where $\mathbf{C}$ is (for $n \times n$ images) the $n^2 \times n^2$ orthogonal DCT matrix and

$$\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{n^2}).$$

Under Tikhonov regularization with regularization parameter λ, $\mathbf{A}_R$ is approximated by $\widetilde{\mathbf{A}}_R$:

$$\widetilde{\mathbf{A}}_R = \mathbf{C}^T\widetilde{\mathbf{\Sigma}}\mathbf{C},$$

where

$$\widetilde{\mathbf{\Sigma}} = \operatorname{diag}(\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_{n^2}) \quad \text{with} \quad \tilde{\sigma}_i = \frac{\sigma_i^2 + \lambda^2}{\sigma_i}.$$

The solution to (3.1) is then computed as

$$\widetilde{\mathbf{A}}_R^{-1}\mathbf{y} = \mathcal{C}^{-1}(\widetilde{\Sigma}^{-1}\mathcal{C}(\mathbf{y})), \tag{3.2}$$

where $\mathcal{C}$ and $\mathcal{C}^{-1}$ denotes the DCT and inverse DCT, respectively. Using fast DCT algorithms, the cost of computing (3.2) is only $O(n^2 \log n)$. The regularization parameter λ can be chosen using a variety of schemes, including discrepancy principle, L-curve, and generalized cross-validation (GCV) [13]. In our work, we use GCV.

The GCV parameter choice method is based on the principle that if a data point is missing, then the remaining data points should predict the missing point well. The regularization parameter λ is chosen to be the minimizer of the GCV function. In our case of Tikhonov regularization on $\mathbf{A}_R$, the GCV function takes the form of

$$G(\lambda) = \frac{\sum_{i=1}^{n^2}\left(\frac{\hat{\mathbf{g}}_i}{\sigma_i^2+\lambda^2}\right)^2}{\sum_{i=1}^{n^2}\left(\frac{1}{\sigma_i^2+\lambda^2}\right)^2}, \quad \text{where} \quad \hat{\mathbf{g}} = \mathcal{C}(\mathbf{g}). \tag{3.3}$$

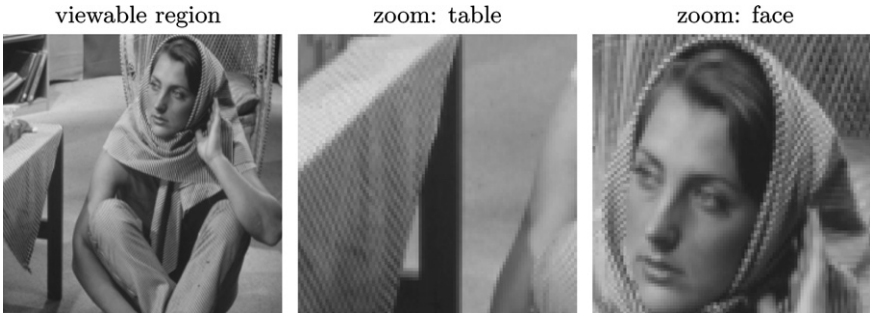


Fig. 4.1. "Barbara" image.

The parameter λ can be obtained by any optimization algorithm on the above function. Details of the implementation of the GCV parameter choice method can be found in [14].

4. Numerical experiments

It is well known that the image deblurring problem requires regularization to stabilize the inversion process when there is noise in $\mathbf{g}$ and/or in $\mathbf{A}$. Note that even if the data $\mathbf{g}$ has no noise (which is highly unlikely in any real problem), because we use only an approximation of the true boundary elements (e.g, with $\mathbf{A}_Z$, $\mathbf{A}_P$, $\mathbf{A}_R$, $\mathbf{A}_A$, or $\mathbf{A}_S$), there is effectively noise in $\mathbf{A}$. For the numerical results reported in this paper we use standard Tikhonov regularization [10,11,13,21],

$$\min_{\mathbf{f}} \left\{ \|\mathbf{g} - \mathbf{A}_X \mathbf{f}\|_2^2 + \lambda \|\mathbf{f}\|_2^2 \right\},$$

where $\mathbf{A}_X$ is one of $\mathbf{A}_Z$, $\mathbf{A}_P$, $\mathbf{A}_R$, $\mathbf{A}_A$, or $\mathbf{A}_S$. Our implementation can be obtained from RestoreTools[1] patched with synthetic boundary conditions modification[2], or Python RestoreTools (PYRET).[3] The following experiments are done with the function HyBR (hybrid bidiagonalization regularization), which implements a modified version of LSQR, in RestoreTools. If the true image is known (as we do in our simulations) HyBR can easily compute Tikhonov solutions with optimal regularization parameters. RestoreTools also facilitates the implementation by providing functions to efficiently implement matrix–vector multiplications.

In our first set of experiments, we use the "Barbara" image (Fig. 4.1) as the main test image. The following four cases are considered:

- Gaussian blur (Section 4.1).
- diagonal motion blur (Section 4.2).
- Gaussian blur with additive Gaussian noise (Section 4.3).
- diagonal motion blur with additive Gaussian noise (Section 4.4).
- DCT based preconditioning with $\mathbf{A}_R$ (Section 4.5).

Results on other images and additional experimental results are also shown in Section 4.6. Note that for display purposes only, pixel values in all of the following figures are clipped to the range $[0, 255]$.

4.1. Gaussian blur

We start with the "Barbara" image, blur it with a Gaussian blur of size 11 with a standard deviation 3 and crop out the central viewable part (Fig. 4.2).

[1] http://www.mathcs.emory.edu/~nagy/RestoreTools.
[2] http://www.mathcs.emory.edu/~yfan/SyntheticBC/SyntheticBcPatch.tgz.
[3] http://www.mathcs.emory.edu/~yfan/PYRET.

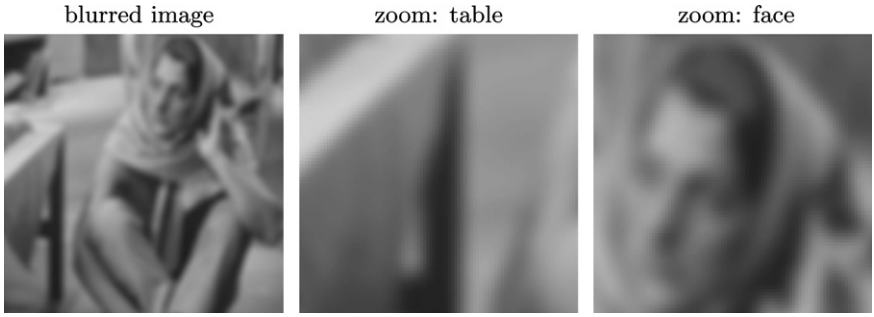
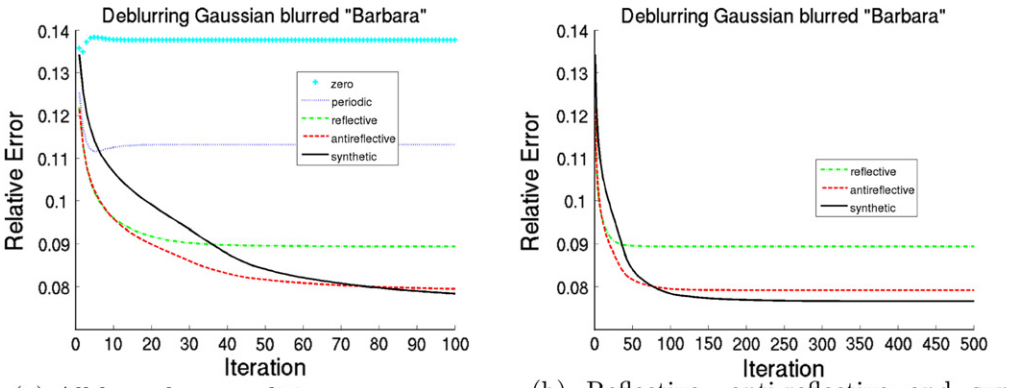


Fig. 4.2. Gaussian blurred "Barbara" image.



(a) All boundary conditions

(b) Reflective, anti-reflective and synthetic boundary conditions

Fig. 4.3. Relative error vs iteration for deblurring Gaussian blurred "Barbara".

To deblur the image, we use the HyBR function with different boundary conditions. The true image is supplied to HyBR to choose the optimal regularization parameters. We run 100 iterations and select the iterates that yield minimum errors. Since the relative errors for reflective, anti-reflective, and synthetic boundary conditions are still decreasing at the 100th iteration, we continue the iterations for these boundary conditions until 500th iteration. The plot of relative errors against iteration is shown in Fig. 4.3. Ideally (when there is no additive noise) with the true boundary conditions, the relative error decreases as the iteration progresses. As can be seen in Fig. 4.3, synthetic and anti-reflective boundary conditions are most faithful to the true boundary conditions, with synthetic performing slightly better than anti-reflective. The corresponding peak signal-to-noise ratios (PSNR),

$$\mathrm{PSNR}(\mathbf{F}, \mathbf{F}_{\mathrm{true}}) = 20 \log_{10} \frac{255}{\mathrm{RMS}(\mathbf{F}, \mathbf{F}_{\mathrm{true}})} = 10 \log_{10} \frac{MN255^2}{\sum_{i,j}[(\mathbf{F})_{i,j} - (\mathbf{F}_{\mathrm{true}})_{i,j}]^2}$$

of the computed reconstructed images are shown in Table 1.

The reconstructed images for the different boundary conditions are shown in Fig. 4.4. From the figure, it is obvious that reconstructions with synthetic boundary conditions contain the least amount of ringing (oscillation) artifacts. The absence of the oscillation is easily seen at the table cloth on the left and the chair behind the woman. Synthetic boundary conditions also give better facial features.

Fig. 4.4. Deblurring results on Gaussian blurred "Barbara" with different boundary conditions.

Table 1
PSNRs of deblurring results on Gaussian blurred "Barbara".

	Blurred image	Zero	Periodic	Reflective	Anti-reflective	Synthetic
PSNR	24.5646	23.8368	25.4884	27.4083	28.4664	28.7532
iteration	–	2	5	167	255	500

4.2. Diagonal motion blur

We repeat the experiment with a diagonal motion blur of size 11; the blurred image is shown in Fig. 4.5. A plot of the relative errors is shown in Fig. 4.6, which clearly illustrates the effectiveness of synthetic boundary conditions compared to other boundary conditions.

Fig. 4.7 shows the computed reconstructions at the point where the iterations reached their smallest error, and the corresponding PSNRs are shown in Table 2. Note that with synthetic boundary conditions we are able to recover the texture of the table cloth and the chair very well, while other boundary conditions either return a blur or texture with severe ringing artifacts. Facial features are also well preserved under synthetic boundary conditions. In terms of PSNRs, synthetic boundary conditions give a significantly higher PSNR than other boundary conditions. Thus, for this particular blurring, our synthetic scheme is most faithful to the true boundary conditions.

4.3. Gaussian blur with additive Gaussian noise

Next, we add 1% Gaussian noise to the Gaussian blurred image and deblur it with different boundary conditions. The noisy blurred image is shown in Fig. 4.8 and the deblurring results are shown in Fig. 4.9.

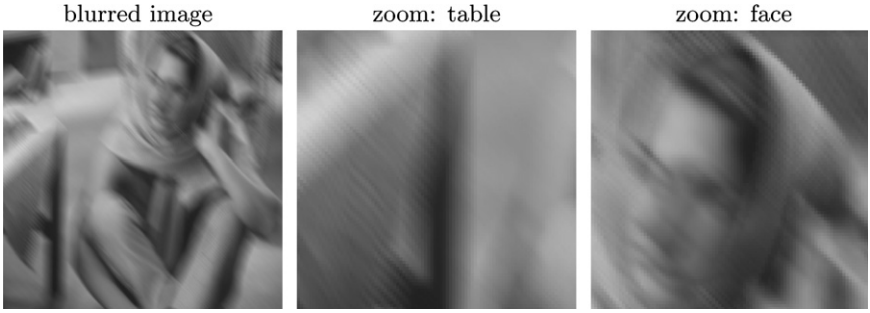


Fig. 4.5. Motion blurred "Barbara" image.

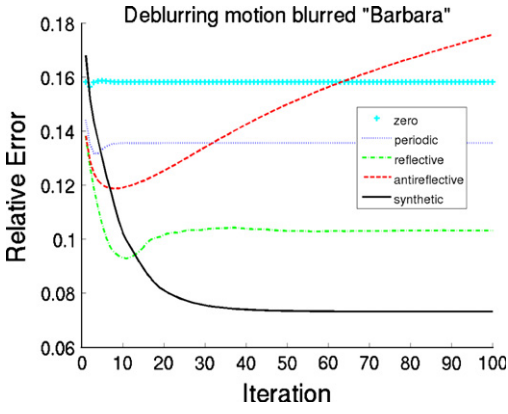


Fig. 4.6. Plot of the deblurring errors vs iteration.

Fig. 4.7. Deblurring results on motion blurred "Barbara" with different boundary conditions.

Table 2
PSNRs of deblurring results on motion blurred "Barbara".

	Blurred image	Zero	Periodic	Reflective	Anti-reflective	Synthetic
PSNR	22.7484	22.5552	24.0441	27.0792	24.9434	29.1441
Iteration	–	2	3	11	8	77

Table 3
PSNRs of deblurring results on noisy Gaussian blurred "Barbara".

	Blurred image	Zero	Periodic	Reflective	Anti-reflective	Synthetic
PSNR	24.5383	23.8341	25.4795	26.9879	27.0608	26.3866
Iteration	–	2	5	23	18	29

Table 4
PSNRs (exlluding outermost 5 pixels) of deblurring results on noisy Gaussian blurred "Barbara".

	Blurred image	Zero	Periodic	Reflective	Anti-reflective	Synthetic
PSNR	24.5383	24.7184	26.0702	27.1973	27.2696	27.3012
Iteration	–	2	6	22	18	25

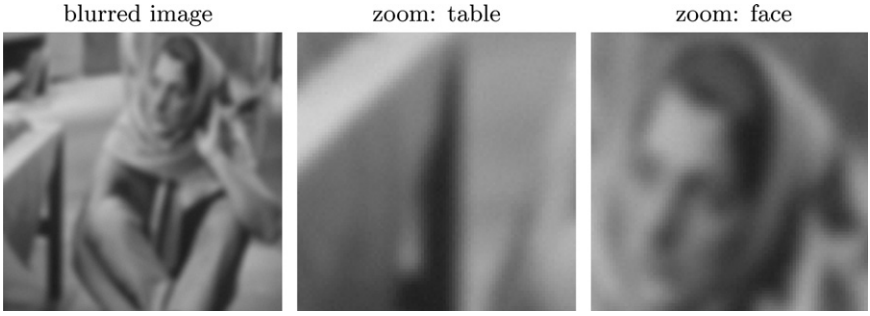


Fig. 4.8. Noisy Gaussian blurred "Barbara" image.

The corresponding PSNRs are shown in Table 3, and the relative error plot against iteration is shown in Fig. 4.10a.

In this case, anti-reflective boundary conditions give the best result, reflective boundary conditions the second best, and synthetic boundary conditions a close third. One may suggest that in the process of obtaining synthetic boundary conditions from the noisy image, noise is taken as image feature and incorrect boundary conditions are obtained. However, we believe this is not true; we applied the synthetic boundary conditions, obtained from the *noisy* blurred image, to deblur the corresponding *noise-free* blurred image, and obtained the very good results shown in Fig. 4.11, with a PSNR of 28.5262 dB. This illustrates that good synthetic boundary conditions can still be obtained from noisy images.

In fact, except for some pixels near the boundary, it is difficult to determine visually if synthetic boundary conditions really perform worse than reflective and anti-reflective boundary conditions. Note that if we exclude the outermost 5 pixels in the calculation of relative errors and PSNRs, anti-reflective and synthetic boundary conditions give very similar results (cf. Table 4 and Fig.4.10b), with slightly better results being obtained with synthetic boundary conditions.

4.4. Diagonal motion blur with additive Gaussian noise

Next, we add 1% Gaussian noise to the motion blurred image and deblur it with different boundary conditions. The noisy blurred image is shown in Fig. 4.12 and the deblurring results are shown in Fig. 4.13. The corresponding PSNRs and a plot of the errors at each iteration are shown in Table 5 and Fig. 4.14, respectively.

Fig. 4.9. Deblurring results on noisy Gaussian blurred "Barbara" with different boundary conditions.

Table 5
PSNRs of deblurring results on noisy motion blurred "Barbara"

	Blurred image	Zero	Periodic	Reflective	Anti-reflective	Synthetic
PSNR	22.7309	22.5497	24.0219	26.4248	24.8341	26.5961
Iteration	–	2	3	9	7	20

Table 6
PSNRs of deblurring results with and without preconditioning.

	Blurred image	Synthetic	Synthetic with preconditioning
PSNR	24.5681	28.7532	29.6790
Iteration	–	500	20

Table 7
PSNRs of the blurred images (Blurred), deblurred images with reflective (Ref), anti-reflective (Antiref) and synthetic (Syn) boundary conditions.

	Gaussian blur				Motion blur			
	Blurred	Ref	Antiref	Syn	Blurred	Ref	Antiref	Syn
Barbara	24.5681	27.4083	28.4664	28.7532	22.7484	27.0792	24.9434	29.1441
Baboon	22.4401	24.3756	25.0833	25.3089	22.0289	26.0322	23.4704	27.8405
Peppers	23.5396	26.5654	27.7779	28.0495	21.4316	25.4560	23.4078	27.5207
Goldhill	24.7255	27.6440	30.3091	29.7711	23.2916	27.9265	25.7929	29.9787
Cameraman	21.2167	26.8596	27.8042	27.8703	20.2303	26.5291	23.8506	29.7191

Table 8
PSNRs of the noisy (1%) blurred images (Blurred), deblurred images with reflective (Ref), anti-reflective (Antiref) and synthetic (Syn) boundary conditions.

	Gaussian blur + 1% Gaussian noise				Motion blur + 1% Gaussian noise			
	Blurred	Ref	Antiref	Syn	Blurred	Ref	Antiref	Syn
Barbara	24.5383	26.9879	27.0608	26.3866	22.7309	26.4248	24.8341	26.5961
Baboon	22.4176	23.5400	23.5663	23.1965	22.0081	25.0278	23.3752	25.2264
Peppers	23.5193	26.3752	26.7754	26.1722	21.4183	25.1881	23.3622	25.7139
Goldhill	24.6953	26.9863	27.2287	26.1062	23.2693	26.8872	25.5519	26.5859
Cameraman	21.2021	23.3848	23.3603	22.8304	20.2189	24.9255	23.3693	24.9763

We observe similar results as in the noise-free case. With synthetic boundary conditions, the texture of the table cloth and chair are restored quite successfully. The facial features are also restored very well. Overall, there are significantly fewer artifacts in the synthetic boundary conditions results compared to the others. In terms of PSNR, synthetic boundary conditions still give the highest PSNR, but its difference from the next best boundary conditions (reflective) is smaller than in the noise-free case.

4.5. Preconditioning

Now we illustrate that preconditioning can significantly accelerate convergence of the iterative method. We only show results for the Gaussian blurred image with synthetic boundary conditions; similar results can be obtained with motion blur. The deblurring results with and without preconditioning are shown in Fig. 4.15, the corresponding PSNRs are shown in Table 6, and the error plots are shown in Fig. 4.16. Recall that without preconditioning, the minimum error is not yet attained even at 500th iteration. With preconditioning, the relative error drops very quickly, attaining its minimum at 20th iteration before increasing a little, and then levels off. In addition, we obtain a higher PSNR and recover more details, e.g. the texture of the chair.

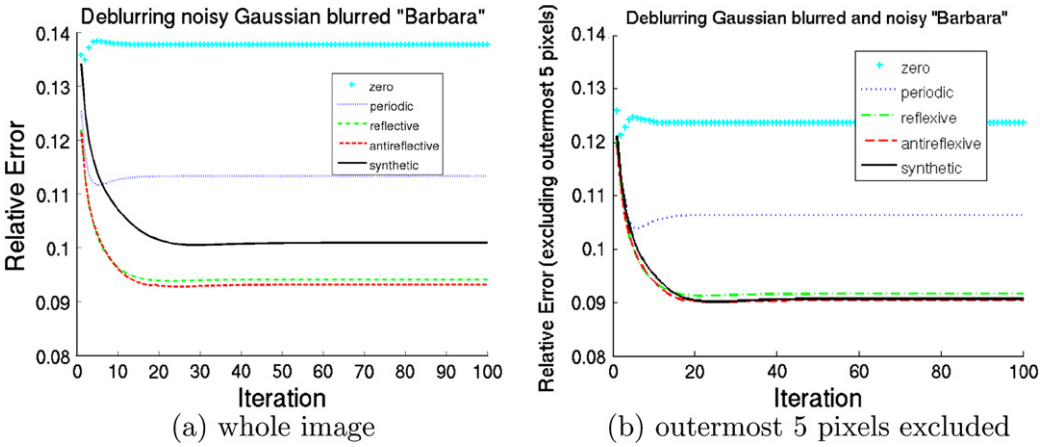


Fig. 4.10. Plot of the deblurring errors vs iteration on noisy Gaussian blurred image. (a) whole image (b) outermost 5 pixels exlluded.

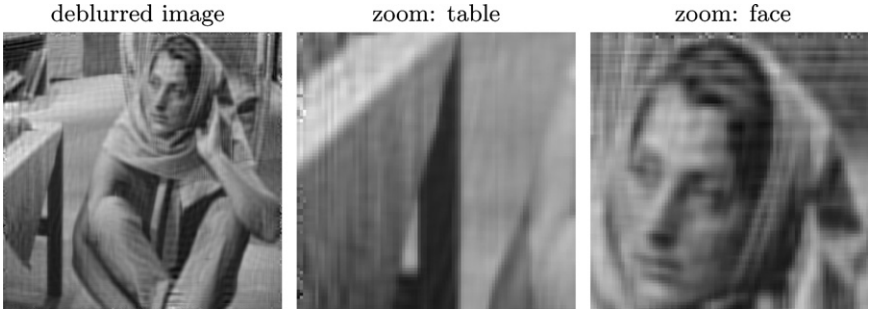


Fig. 4.11. Deblurring result of Gaussian blurred "Barbara" with the synthetic boundary conditions obtained from the blurred and noisy counterpart. Its PSNR to the original image is 28.5 dB.

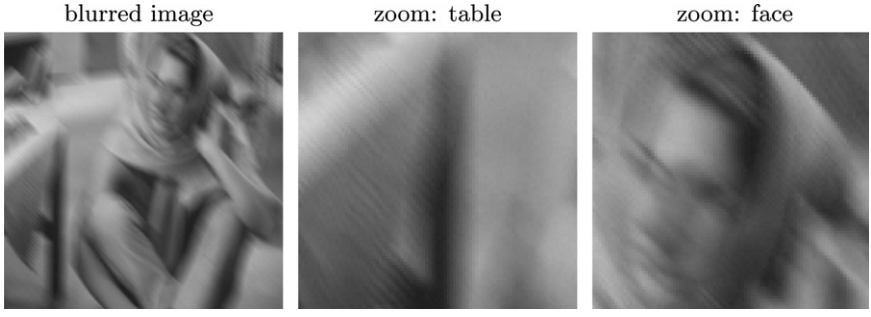


Fig. 4.12. Noisy motion blurred "Barbara" image.

4.6. Other images and additional experiments

We repeated all of the experiments on several other standard test images (see, e.g., the MATLAB Image Processing Toolbox) with reflective, anti-reflective and synthetic boundary conditions respectively. The results are shown in Tables 7 and 8. Synthetic boundary conditions almost always give the highest PSNRs. Occasionally, synthetic boundary conditions give slightly lower PSNRs than anti-reflective boundary conditions, such as in the case of deblurring the Gaussian blurred "Goldhill" image. But

Fig. 4.13. Deblurring results on noisy motion blurred "Barbara" with different boundary conditions.

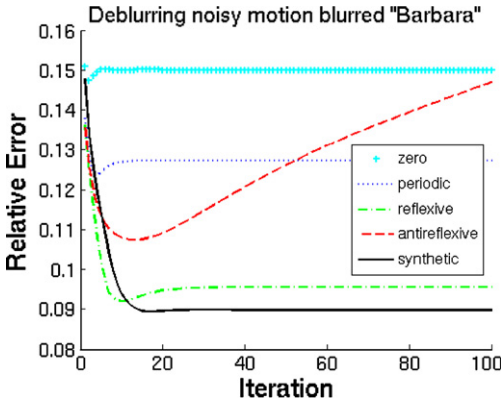


Fig. 4.14. Plot of the deblurring errors vs iteration on noisy motion blurred "Barbara".

Fig. 4.15. Deblurring results on Gaussian blurred "Barbara" with synthetic boundary conditions. The first row is obtained without preconditioning at the 500th iteration; the second row is obtained with preconditioning at the 20th iteration.

Table 9
PSNRs of the slightly noisy (0.1%) blurred images (Blurred), deblurred images with reflective (Ref), anti-reflective (Antiref) and synthetic (Syn) boundary conditions.

	Gaussian blur + 0.1% Gaussian noise				Motion blur + 0.1% Gaussian noise			
	Blurred	Ref	Antiref	Syn	Blurred	Ref	Antiref	Syn
Barbara	24.5644	27.3972	28.3176	28.3448	22.7321	26.4194	24.8416	26.5773
Baboon	22.0290	26.0191	23.4691	27.7776	22.0291	26.0151	23.4695	27.7875
Peppers	23.5395	26.5659	27.7148	27.8312	21.4312	25.4549	23.4084	27.4904
Goldhill	24.7251	27.6196	29.6363	28.9025	23.2913	27.9057	25.7940	29.8515
Cameraman	21.2165	25.7648	25.9675	25.6989	20.2301	26.5009	28.8533	29.5292

in these cases synthetic boundary conditions still produce fewer ringing artifacts than anti-reflective boundary conditions; see, for example, Fig. 4.17.

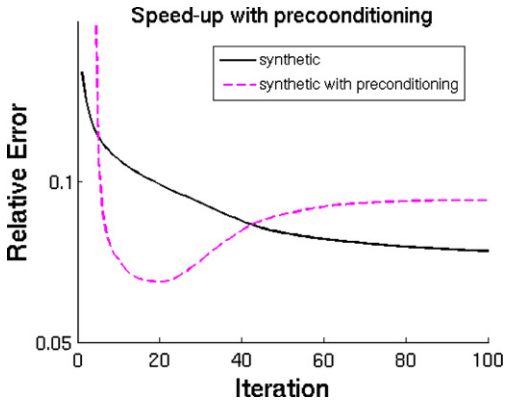


Fig. 4.16. Plot of the deblurring errors with and without preconditioning.

Table 10
PSNRs of the slightly blurred (PSF size: 5×5) images (Blurred), deblurred images with reflective (Ref), anti-reflective (Antiref) and synthetic (Syn) boundary conditions.

	Gaussian blur				Motion blur			
	Blurred	Ref	Antiref	Syn	Blurred	Ref	Antiref	Syn
Barbara	28.3808	32.6027	33.5761	32.1339	26.5846	32.9626	30.4176	34.1026
Baboon	24.4705	28.2983	29.3215	28.4046	23.9554	29.0244	26.3301	30.3743
Peppers	28.6461	33.5144	34.8624	32.2711	26.3432	31.9760	30.1423	32.1449
Goldhill	28.5885	34.8708	36.7407	34.4301	27.0294	33.7078	30.8930	35.1045
Cameraman	24.3743	32.0713	32.5470	31.6834	23.0170	30.2952	27.4850	33.3327

In Table 9, we show the results when the noise level is only 0.1%. The results are similar to the noise-free case. Synthetic boundary conditions give the highest PSNR in all cases except on the noisy Gaussian blurred Goldhill image.

We show in Table 10 the results when the PSF size is only 5×5. With a smaller PSF, the border region $\mathcal{B}$ in Fig. 2.1 is narrower and thus the effect of synthetic boundary conditions in continuing edge direction and texture is less significant. This is clearly illustrated for the Gaussian blurred images, where we see that although the reflective and anti-reflective boundary conditions give slightly better results than synthetic boundary conditions, the difference in all cases is small. For motion blurred images, synthetic boundary conditions result in highest PSNRs even with a narrow border. This again demonstrates the strength of synthetic boundary conditions in deblurring motion blurred images.

5. Conclusions

We have introduced a new approach to choosing boundary conditions for imaging applications. We described the approach, which we call *synthetic boundary conditions*, in the context of image deblurring, and compared its linear algebraic structure, as well as its effectiveness to previously proposed boundary conditions. All four previously proposed boundary conditions (zero, periodic, reflective and anti-reflective) fail to continue important image structures like edge directions and texture outside the viewable region. On the other hand, our synthetic approach can continue these image structures. Extensive numerical experiments illustrated that synthetic boundary conditions typically allow for (sometimes significantly) better image reconstructions than other boundary conditions. In the (rare) situations when other boundary conditions performed better than our synthetic approach, the difference was minimal, and visually one could argue that the reconstructions with synthetic boundary conditions had fewer artifacts. The linear algebraic structure of the new boundary condition allows for efficient implementation of iterative image deblurring algorithms, and construction of effective preconditioners.

Fig. 4.17. Deblurring results on noisy Gaussian blurred "Goldhill" with anti-reflective and synthetic boundary conditions.

References

[1] A. Aricò, M. Donatelli, S. Serra-Capizzano, Spectral analysis of the anti-reflective algebra,Linear Algebra Appl. 428 (2008) 657–675.
[2] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image in painting, Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, Addison-Wesley Publishing Co., New York, NY, USA, 2000, pp. 417–424.
[3] R.H. Chan, X.Q. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, 2007.
[4] P.J. Davis, Circulant Matrices, Wiley, New York, 1979.
[5] F. Di Benedetto, C. Estatico, S. Serra-Capizzano, Superoptimal preconditioned conjugate gradient iteration for image deblurring, SIAM J. Sci. Comput. 26 (2005) 1012.

[6] M. Donatelli, C. Estatico, A. Martinelli, S. Serra-Capizzano, Improved image deblurring with anti-reflective boundary conditions and re-blurring, Inverse Problems 22 (6) (2006) 2035.
[7] M. Donatelli, C. Estatico, J. Nagy, L. Perrone, S. Serra-Capizzano, Anti-reflective boundary conditions and fast 2D deblurring models, in: F. Luk (Ed.), Proceeding to SPIEs 48th Annual Meeting, San Diego, CA USA, vol. 5205, 2003, pp. 380–389.
[8] A.A. Efros, W.T. Freeman, Image quilting for texture synthesis and transfer, in: Proceedings of SIGGRAPH 2001, Los Angeles, CA, 2001, pp. 341–346.
[9] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling, in: International Conference on Computer Vision, 1999, pp. 1033–1038.
[10] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer Academic Publishers, Dordrecht, 2000.
[11] C.W. Groetsch, The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind, Pitman, Boston, 1984.
[12] M. Hanke, J. Nagy, R. Plemmons, Preconditioned iterative regularization for ill-posed problems, Numerical Linear Algebra: Proceedings of the Conference in Numerical Linear Algebra and Scientific Computation, Kent (Ohio), USA, March 13–14, 1992, Walter de Gruyter, 1993, pp. 141.
[13] P.C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1998.
[14] P.C. Hansen, J.G. Nagy, D.P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.
[15] R.L. Lagendijk, J. Biemond, Iterative Identification and Restoration of Images, Kluwer Academic Publishers, Boston/Dordrecht/London, 1991.
[16] C.F. Van Loan, Computational Frameworks for the Fast Fourier Transform, SIAM, Philadelphia, 1992.
[17] J.G. Nagy, K. Palmer, L. Perrone, Iterative methods for image deblurring: a Matlab object-oriented approach, Numer. Algorithms 36 (1) (2004) 73–93.
[18] M.K. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, Oxford, UK, 2004.
[19] M.K. Ng, R.H. Chan, W.C. Tang, A fast algorithm for deblurring models with Neumann boundary conditions, SIAM J. Sci. Comput. 21 (1999) 851–866.
[20] S. Serra-Capizzano, A note on antireflective boundary conditions and fast deblurring models, SIAM J. Sci. Comput. 25 (4) (2003) 1307–1325.
[21] C.R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.